CLAIMS

1. A method of sending a message from a first common Object Request Broker to a second common Object Request Broker operating in a distributed object oriented environment, said method comprising:

5   determining whether the message is to be fragmented in two or more sub-messages;

initiating construction of a sub-message when said determining determines that said message is to be sent in two or more sub-messages;

initializing an offset-variable to zero when said determining determines that said message is to be fragmented into two or more sub-messages;

10   determining whether there is a need to know the position of a byte of the sub-message with respect to the message;

reading the offset-variable when said determining determines that there is a need to know the position of a byte of the sub-message with respect to the message;

completing construction of the sub-message based on the offset-variable;

15   updating the offset-variable; and

sending a constructed sub-message from the first common Object Request Broker to a second common Object Request Broker.

2. A method as recited in claim 1,

wherein said sub-message has a header that includes one or more bytes, and

20   wherein said updating of the offset-variable comprises:

subtracting the length of the header of the sub-message from the length of another sub-message that was constructed immediately prior to construction of the sub-message.

3. A method as recited in claim 2, where said updating further comprises:

14

updating the offset-variable by adding the result of said subtracting to the value of the offset-variable;

4. A method as recited in claim 3,

wherein the message is fragmented into N sub-messages, sub-message 0 to sub-message N, where N is a positive integer, and

wherein the first sub-message has a header with the same number of bytes as the header of the message.

5. A method as recited in claim 4, wherein the updating is performed based on the summation $(L_{i-1} - H_i)$ taken from $i = 1$ to $i = N-1$, where L_i is the length of the sub-message i and H_i is the length header of the sub-message i.

6. A method as recited in claim 5, wherein at least two of the headers of the sub-messages 1 to N can be of different lengths.

7. A method as recited in claim 1, wherein the method further comprises:

obtaining a remote object; and

invoking a method associated with the object.

8. A method as recited in claim 7,

wherein said obtaining of the remote object and said invoking of a method associated with the object is performed by a client operating in the distributed object oriented environment, and

wherein the first Object Request Broker creates a request and marshals in appropriate parameters.

9. A computing system operating in a distributed object oriented environment, said computing system comprising:

a first common Object Request Broker operating to send a message to a second common Object Request Broker, said message being fragmented by the first common Object Request Broker into two or more sub-messages in a sequence, and

wherein the position of a byte of a sub-message with respect to the message can be determined based on a offset-variable by subtracting the length of the header of the sub-message from the length of another sub-message immediately preceding the sub-message, and then adding the result of the subtraction to the value of the offset-variable.

5   10. A computing system as recited in claim 7, wherein the message is fragmented into N sub-messages, sub-message 1 to sub-message N, where N is a positive integer, and

wherein the first sub-message has a header with the same number of bytes as the header of the message.

11. A computing system as recited in claim 8, wherein at least two of the headers of the sub-
10   messages 1 to N can be of different lengths.

12. A computing system as recited in claim 8, wherein at least two of the sub-messages have data portions that are of different sizes.

13. A method of sending a message from a first common Object Request Broker to a second common Object Request Broker operating in a distributed object oriented environment, said
15   method comprising:

initiating construction of a sub-message when said determining determines that said message is to be sent in two or more sub-messages;

determining whether there is a need to know the position of a byte of the sub-message with respect to the message;

20   reading the offset-variable when said determining determines that there is a need to know the position of a byte of the sub-message with respect to the message;

completing construction of the sub-message based on the offset-variable; and

updating the offset-variable by subtracting the length of the header of the sub-message from the length of another sub-message that was constructed immediately prior to construction
25   of the sub-message and adding the result of the subtraction to the value of the offset-variable.

14. A method as recited in claim 13, wherein the first sub-message has a header with the same number of bytes as the header of the message.

16

15. A method as recited in claim 13, wherein at least two of the headers of the sub-messages can be of different lengths.

16. A method as recited in claim 13, wherein at least two of the sub-messages have data portions that are respectively of different sizes.

5   17. A computer readable media including computer program code for sending a message from a first common Object Request Broker to a second common Object Request Broker operating in a distributed object oriented environment, said method comprising:

computer program code for determining whether the message is to be fragmented in two or more sub-messages;

10   computer program code for initiating construction of a sub-message when said determining determines that said message is to be sent in two or more sub-messages;

computer program code for initializing an offset-variable to zero when said determining determines that said message is to be fragmented in two or more sub-messages;

computer program code for determining whether there is a need to know the position
15   of a byte of the sub-message with respect to the message;

computer program code for reading the offset-variable when said determining determines that there is a need to know the position of a byte of the sub-message with respect to the message;

computer program code for completing construction of the sub-message based on the
20   offset-variable;

computer program code for updating the offset-variable; and

computer program code for sending a constructed sub-message from the first common Object Request Broker to a second common Object Request Broker.

18. A computer readable media as recited in claim 17,

25   wherein said sub-message has a header that includes one or more bytes, and

wherein said computer program code for updating the offset variable comprises:

17

computer program code for subtracting the length of the header of the sub-message from the length of another sub-message that was constructed immediately prior to construction of the sub-message.

19. A computer readable media as recited in claim 17, wherein said computer program code for updating further comprises:

computer program code for updating the offset-variable by adding the result of said subtracting to the value of the offset-variable.

20. A computer readable media as recited in claim 17, wherein at least two of the headers of the sub-messages can be of different lengths.

21. A computer readable media as recited in claim 17, wherein at least two of the data portions of the sub-messages can be of different lengths.